EMBEDDED OPEN SOURCE SUMMIT

EMBEDDED LINUX CONFERENCE

Software Optimizations for Early Boot on Time Critical Applications

2024-04-18, Seattle

Aashvij Shenai

Soumya Tripathy

Texas Instruments

About us: TI Processors and Open source

Decades of contribution and collaboration

Ingrained culture to give back to the community

Upstream FIRST!

Focus on long term, sustainable and quality products

Upstream and opensource ecosystem in device architecture

Upstream FIRST mentality!

Introduction to the Speakers

Aashvij Shenai, *Senior Software Engineer, Texas Instruments*

Aashvij primarily works in the Systems domain and is presently handling Key Performance Indicators (KPIs) in boot sequence. His hobbies include automating things that do not need it and creating board-games.

Soumya Tripathy, *Lead Engineer, Texas Instruments*

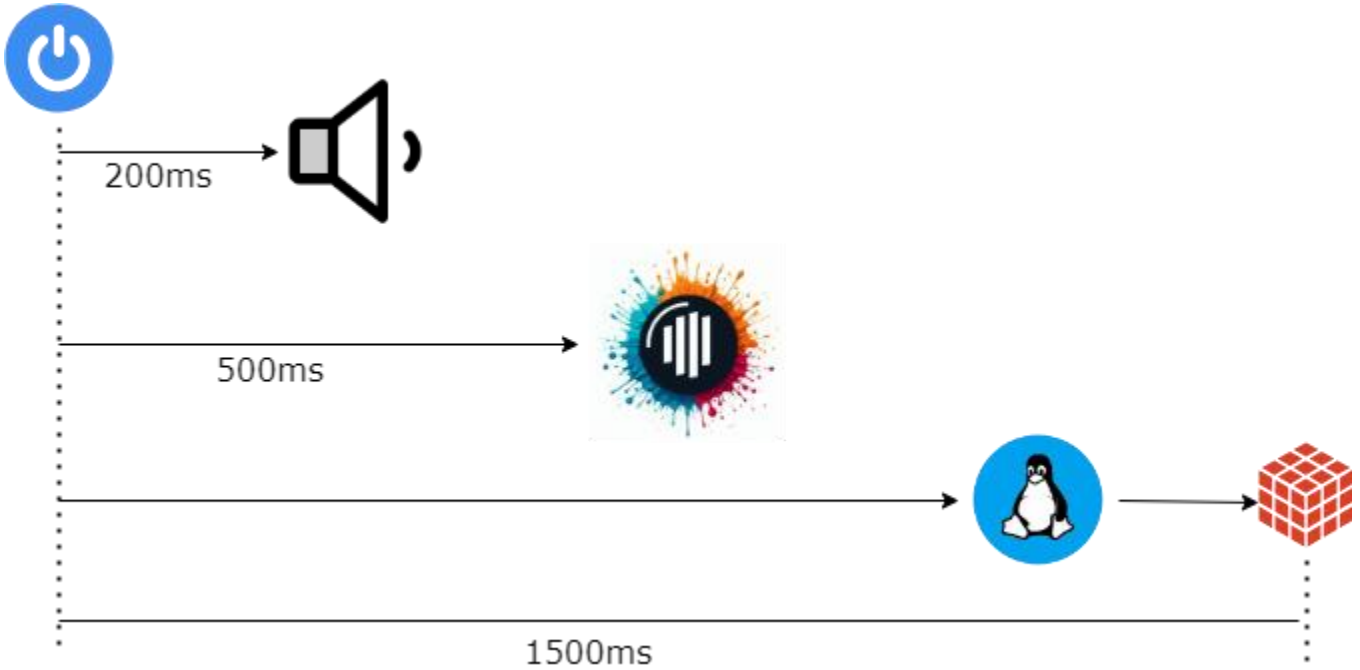
Soumya has been working with TI for 2 years with contributions and expertise in the field of bootloader, flash devices and display controller for the Sitara family of processors. Prior to TI, he worked as a firmware developer at Robert Bosch. He made contributions to firmware development of Servo drives by Bosch Rexroth related to the field of Industrial communication and factory automation products.

Overview

- Problem Statement
- Existing Sequence
- Optimizations
- Thoughts, surprises

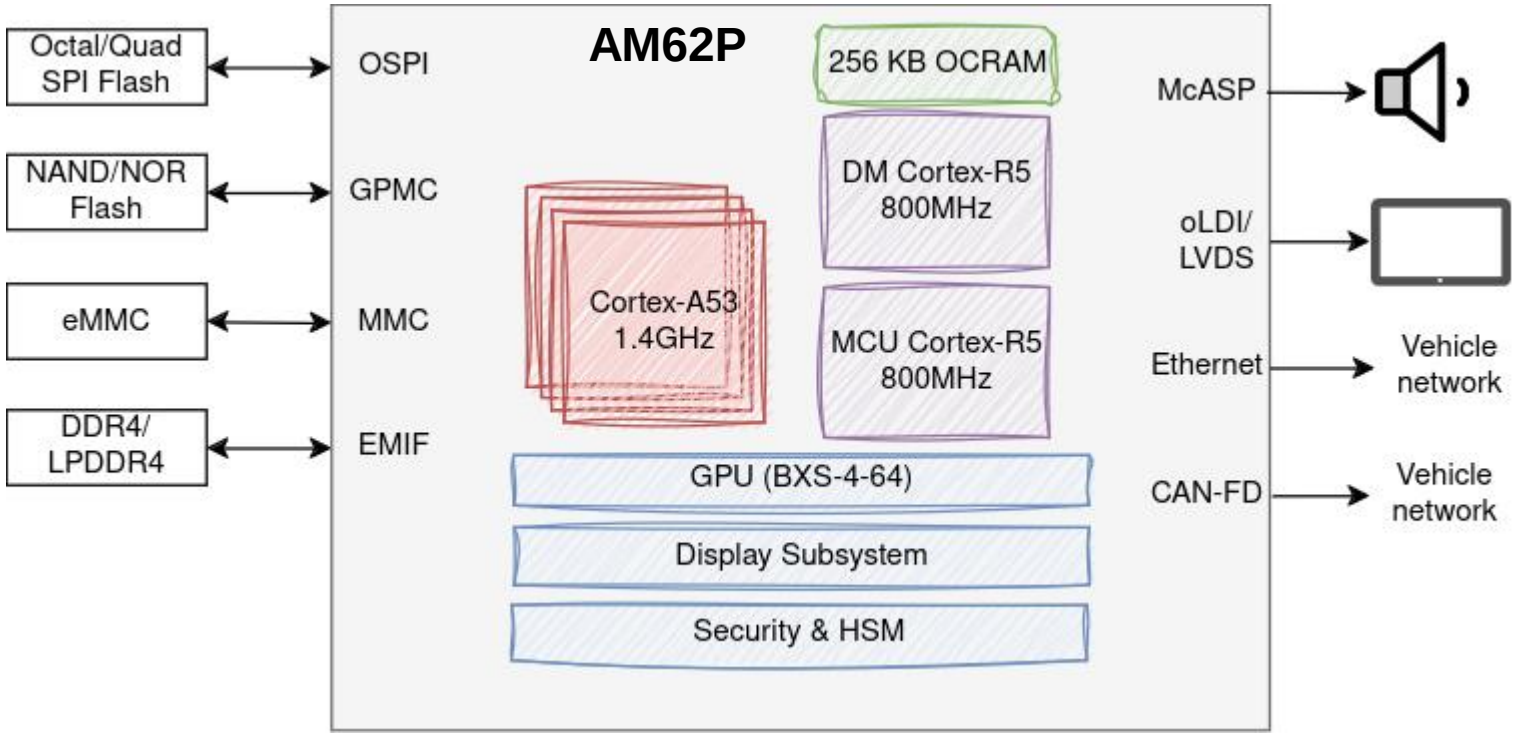
Problem Statement

How do we handle multicore applications in the early boot scenario?

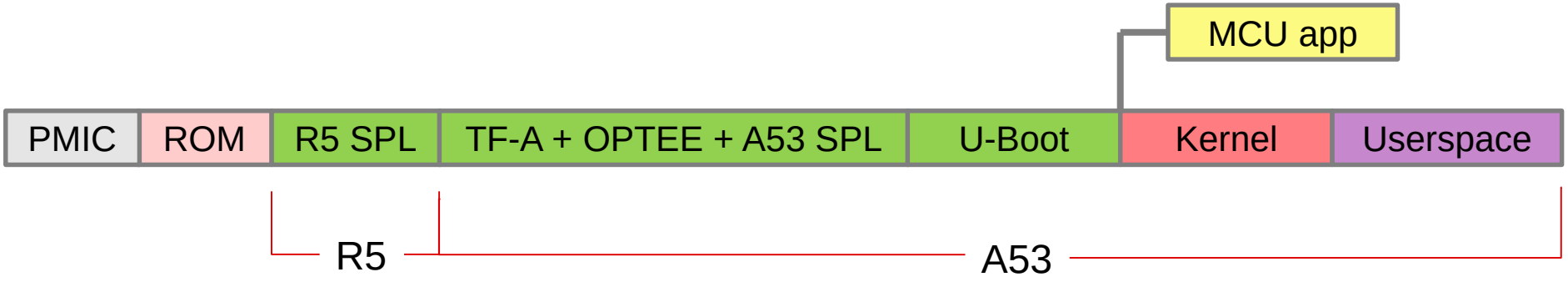


Existing Sequence

System


Octal/Quad SPI Flash
NAND/NOR Flash
eMMC
DDR4/ LPDDR4
OSPI
GPMC
MMC
EMIF
AM62P
256 KB OCRAM
DM Cortex-R5 800MHz
Cortex-A53 1.4GHz
MCU Cortex-R5 800MHz
GPU (BXS-4-64)
Display Subsystem
Security & HSM
McASP
oLDI/ LVDS
Ethernet
Vehicle network
CAN-FD
Vehicle network

Boot Sequence



Boot Sequence - I

PMIC

- Handles voltage regulation, power monitoring and sequencing of the SoC

ROM

- Configures device resources (PLLs, peripherals, pins) to support boot
- Jumps to bootloader

Boot Sequence - II

R5 SPL

- The first stage of the bootloader, it is small enough to fit in the OCRAM. It configures and initializes peripherals and most importantly, DDR.
- It loads, TF-A, OPTEE and U-boot proper into the DDR and then jumps to TF-A

TF-A + OPTEE + A53 SPL

- TF-A - Provides a reference trusted code base for all Armv8 cores on ARMv8.
- OPTEE - TF-A installs the secure world software (OP-TEE) and passes execution on to A53 SPL
- A53 SPL – Linux friendly bootloader that loads U-Boot proper

Boot Sequence - III

U-Boot

- Offers a flexible way to load and start the Linux Kernel and provides a minimal set of tools to interact with the board's hardware via a command line interface.
- It runs from DDR, initializing additional hardware devices (network, USB). Then, it loads and prepares the device tree

Kernel

- Goes through a multithreaded initialization to load system features as well features requested by the user

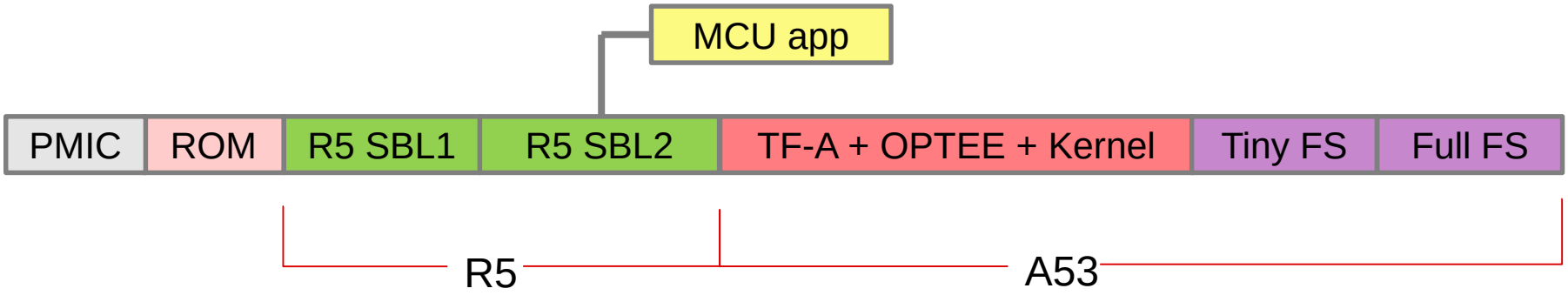
Boot Sequence - IV

Userspace

- Part of the OS where a user's applications are run, isolated from the kernel.
- Provides an environment for applications to interact with the hardware and other system resources

Optimized Sequence

Boot Sequence



Resolve and standardize early access to hardware for automotive industry with Linux – Khasim Syed Mohammed

Boot Sequence - I

PMIC

- Handles voltage regulation, power monitoring and sequencing of the SoC, but it also … *does it faster*

ROM

- Initializes boot peripherals and jumps to the next stage
- Implements an algorithm to switch OSPI-NOR flash into 8D-8D-8D data transfer mode at 166MHz to max throughput

Boot Sequence - II

R5 SBL1

- The first stage of the bootloader, it is small enough to fit in the OCRAM. It configures and initializes peripherals and most importantly, DDR.
- It does not reinitialize the boot media drivers that have been successfully configured by ROM

R5 SBL2

- This multithreaded stage loads applications for all heterogenous cores (Security M4, MCU R5, A53) while setting up the Device Manager in tandem
- Supports loading of a flat binary image of TF-A, OPTEE and Kernel
- Uses DMA to quickly load images

Boot Sequence - III

MCU app

- In the multithreaded context, audio application is run while display buffers are being prepared.
- Once early chime is done, switch to splash screen

Boot Sequence - IV

TF-A + OPTEE + Kernel

CONFIG	Kernel size reduction	Time reduction
RODATA_FULL_DEFAULT_ENABLED	-	144ms
PTP_1588_CLOCK	69.6KB	64ms
DEBUG_FS	335.9KB	48ms
MMC	137.2KB	32ms
VFAT_FS	131.1KB	32ms
SPI	208.9KB	16ms

TI Optimized config - git.ti.com

Boot Sequence - V

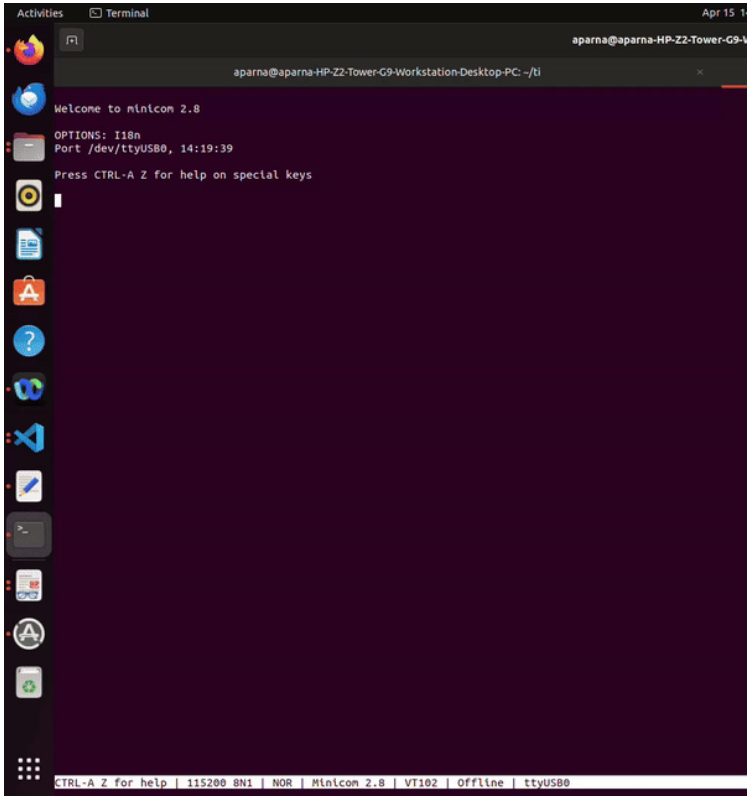
Tiny FS

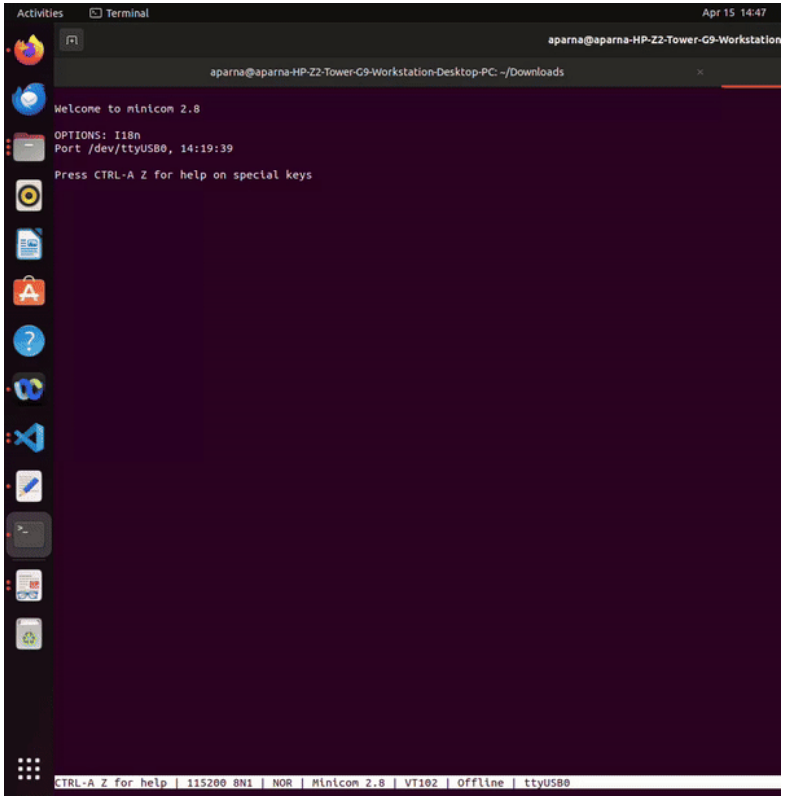
- Init manager – sysvinit
- What services are required?
- Filesystems – UBIFS, SquashFS

Boot Sequence - VI

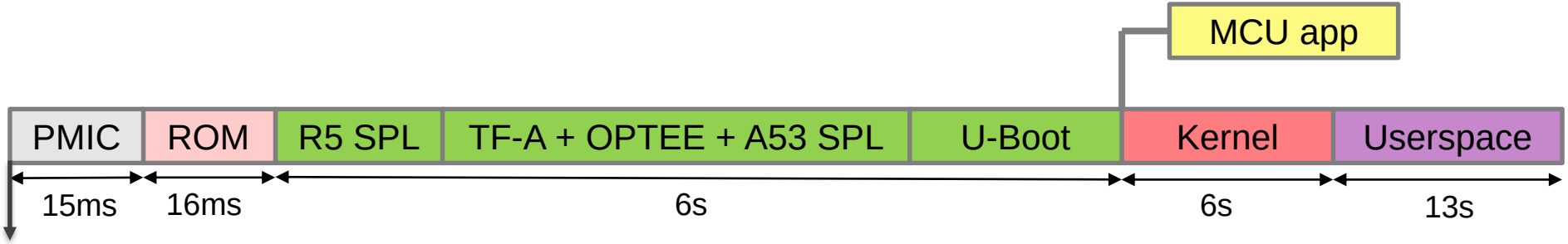
Filesystem	Size (MB)	Increase in mount time
EXT4	9.8	-
UBIFS min-io:2K, leb:126KB, max-leb:2K	7.0	+48ms
SquashFS B:1M, C:GZIP	3.7	+77ms
SquashFS B:1M, C:XZ	2.5	+103ms
SquashFS B:1M, C:LZ4-HC	5.0	+89ms

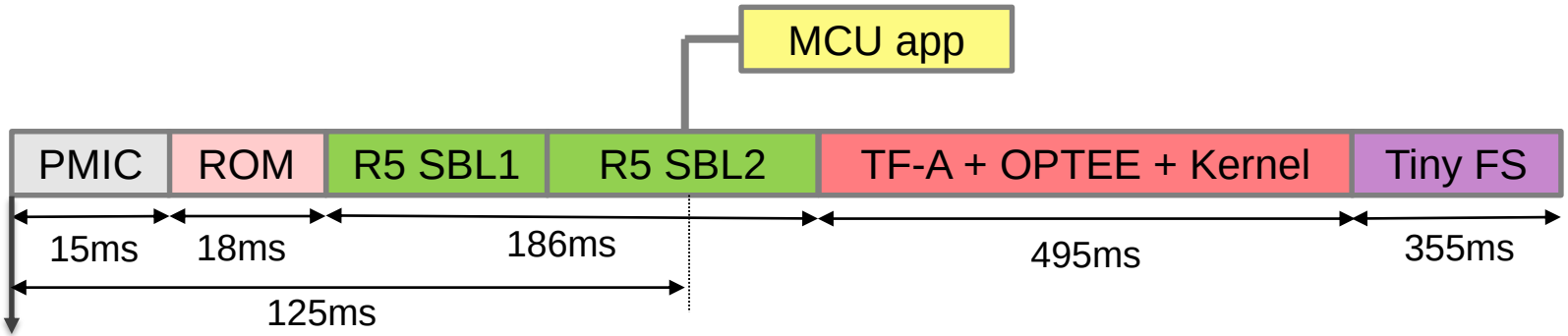
Boot Sequence - VII





Boot Sequence - VIII





Thoughts and surprises

- How do we get an RTOS bootloader out of U-Boot?
- How do we keep a barebones filesystem in mind when building Yocto recipes?

- Why did UBIFS fail against EXT4?

References

- Linux Boot Time Optimizations on DRA7xx Devices
- Linux Boot Time Optimizations for i.MX8M Family

Recreating the environment

- Creating the UBIFS:
  mkfs.ubifs <source> -m 2048 –e129024 –c 2048 –o <destination>
- Creating the SquashFS:
  mksquashfs <source> <destination> -b <block size> -comp <compression algorithm>

- 4.1.9. Boot Time Optimizations — Processor SDK AM62Px Documentation

Q&A

- Contact Information:
  - Aashvij Shenai <a-shenai@ti.com>
  - Soumya Tripathy <s-tripathy@ti.com>

Learn more about TI products

- https://www.ti.com/linux
- https://www.ti.com/processors
- https://www.ti.com/edgeai

Why choose TI MCUs and processors?

Scalability

Our products offer scalable performance that can adapt and grow as the needs of your customers evolve.

Efficiency

We design products that extend battery life, maximize performance for every watt expended, and unlock the highest levels of system efficiency.

Affordability

We strive to make innovation accessible to all by creating cost-effective products that feature state-of-the-art technology and package designs.

Availability

Our investment in internal manufacturing capacity provides greater assurance of supply, supporting your growth for decades to come.

Systemd - logs